What I have covered in class so far:

Program. Function.

Machine language, assembly language and high level languages

Source code, Object Code, Executable code. Editor, Compiler, linker and loader.

Syntax, Semantics, runtime and logical errors

Algorithm- Step-by-step problem-solving process. Solution achieved in finite amount of time

Constant, variable. Data types (set of values together with a set of operations)

 A variable should be declared with a name and type before using. First time you place a value in a variable is called initializing. Remember, **it never is zero** when you declare a variable. If you want it to be 0, you must place 0 in the variable.

C++ data types fall into three categories: (1) Simple data type (2) Structured data type (3) Pointers

We will work with simple data types for several weeks. Such as int, float (6 or 7 significant digits-needs four bytes), double (15 significant digits –needs 8 bytes), char, and bool.

Memory, address, each cell can contain a programming instruction or data

EBCDIC, ASCII, Unicode

Preprocessor directives begin with # and are processed by the preprocessor



Two popular approaches to programming design: Structured and Object-oriented

cin>> cout<<

Today we will cover:

* C++ arithmetic operators:
	+ + addition
	+ - subtraction
	+ \* multiplication
	+ / division
	+ % modulus (or remainder) operator
* +, -, \*, and / can be used with integral and floating-point data types
* Use % only with integral data types

Order of precedence

* All operations inside of () are evaluated first
* \*, /, and % are at the same level of precedence and are evaluated next
* + and – have the same level of precedence and are evaluated last
* When operators are on the same level
	+ Performed from left to right (associativity)
* 3 \* 7 - 6 + 2 \* 5 / 4 + 6 means

(((3 \* 7) – 6) + ((2 \* 5) / 4 )) + 6

We will run a program that will demonstrate this.

= assignment symbol in c++ .

== equal to test in C++